FRVT Quality Assessment - Specific Image Defect Detection

An ongoing evaluation of software that checks for quality problems in face images

API and Concept Document – 2022-07-01

1. FRVT 1:1 Verification	**2. FRVT 1:N** Search Performance	**3. FRVT Morph** Morphed Photo Detection	**4A FRVT** QA Image Quality Scalar Summary	**4B FRVT QA** Specific Image Defect Detection	**5. FRVT Attack** Presentation Attack Detection
2017...	**2018...**	**2018...**	**2019...**	**2022 Q3 ...**	**2022 Q3 ...**

THIS DOCUMENT IS OPEN FOR PUBLIC COMMENT UNTIL 2022-08-18

PLEASE SUBMIT COMMENTS + QUESTIONS TO FRVT@NIST.GOV

For further updates and links see the FRVT Quality page
https://pages.nist.gov/frvt/html/frvt_quality.html

Contents

- How to participate
- Role, context, scope
  - Relationship to ISO/IEC 29794-5 now under development
- API
- Detailed description of quality measurements

FRVT SIDD: How a developer can participate

- Read this document
- Read the API
- Read the participation agreement; agree to it, sign it, scan it to PDF.
- Implement one or more image quality methods enumerated in the API, and described below
- Download the FRVT quality validation package; compile, link, run, check output
- tar (or zip) the combined software and validation output; sign and encrypt the tar.gz
- Email frvt@nist.gov with
  - A download link to the encrypted package tar.gz.gpg
  - A PDF of the scan of the paper participation agreement
    - Do not mail a paper copy for this track of FRVT
  - Your public key (that was used to sign the tar.gz file)
- Subscribe to FRVT news
- …
- Consult https://pages.nist.gov/frvt/html/frvt_quality.html

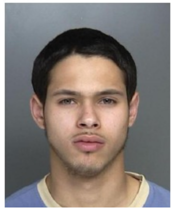
Timeline:

1. 2022-07-05: First draft
2. 2022-08-18: Comments due
3. 2022-09-19: Final API published
4. 2022-09-19: Implementations can be submitted

FRVT Quality tracks

TRACK 4A
Q Summaries

Website



SCALAR: Q = 98

DECISION: Y, Accept

BOX 0. QUALITY BENCHMARK
- One "visa – border" dataset
- Wild dataset no longer in use

TRACK 4B
Q Diagnostics

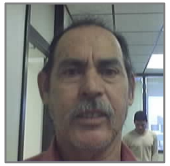
Website

BOX 1. QUALITY BENCHMARK
- Concept presented at the Nov Q Workshop
  - Publish 2022-07
  - Developer comment
- Algorithms to NIST 2022-08
- Align with ISO/IEC 29794-5

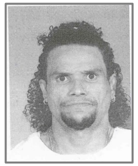
BOX 2. IMAGING VARIABLES THAT INFLUENCE ACCURACY
- Illumination adequacy + uniformity
- Exposure
- Focus, blur
- Resolution / Sp. Sampling Rate
- …

BOX 3. SUBJECT VARIABLES THAT INFLUENCE ACCURACY
- Head orientation (R, P, Y)
- Expression neutrality
- Sunglasses, face masks
- Motion blur
- No, or additional, faces
- …

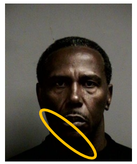
Examples

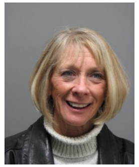

Two People

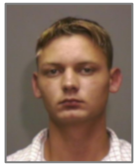
No People

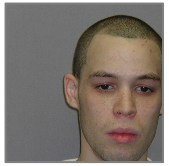

Noise

Over-exposure


Under-exposure


Mouth open


Loss of resolution


Cropped Misplaced

Non-frontal

FRVT SIDD: Two roles

Support Quality Algorithm Development

- Assess capability of algorithms to quantify specific properties of faces in images that are associated with degraded face recognition performance
  - e.g. blur, non-frontal view

Support ISO/IEC 29794-5 Face Image Quality

- Discussion in 2022-07-07, 2023-01 ...
  - In ISO/IEC JTC 1 / SC 37 Working Group 3
  - New drafts a few weeks after the meetings
  - Contact patrick DOT grother AT nist DOT gov on how to participate
- The standard is defining
  - A set of specific tests (image processing operations) to be performed on image; test results can be used to give actionable feedback to a photographer or subject
  - Numeric values (penalties) and data-types for the results of tests, and
  - An interpretable interoperable container for the results
- FRVT will support development by
  - Testing whether a 29794-5 quality method expresses something that has influence on face recognition accuracy
  - Inform how to penalize a quality problem e.g. how should underexposure, or yaw angle, be penalized

- The draft of 29794-5 may include quantities not tested here.
  - Whether those quantities should be in the standard is beyond our scope here.
- There may be quantities tested here that do not appear in the draft of 29794-5
  - For example, the number of faces in the image.
- The numerical expression of a quantity here may not be the same as that given in the standard.
  - The standard seeks to measure certain quantities and then encode them as integers on [0,100] (so that quality thresholds could be applied).
  - The quality values penalize various attributes, often using a sigmoid function to smoothly increase of decrease the penalty
  - Our purpose is to test capability

Criteria for inclusion a quantity in FRVT SIDD

Required property of the quality metric

- Quantity should be related to recognition outcomes
  - Example YES: Resolution
  - Example NO: Eyes closed
- Quantity must be measurable from an image
  - Example YES: Yaw angle
  - Example NO: Exposure duration
- Quantity must have a quantitative definition
  - Example YES: Mouth openness
  - Example NO: Expression neutrality
- Quantity could be quickly remedied in an operational setting
  - Example YES: Sun glasses present
  - Example NO: Signal to noise ratio
- Quantity should be capable of being measured on sequestered datasets (at NIST)
  - To separate developer-training from our testing

Properties not considered

- **Eye openness** is not considered in this document because it has little known impact on false negative outcomes, and varies widely with ethnicity
- **Compression** - is widely abused. A compression ratio can be estimated from the image. Poor values for JPEG are known; there's nothing for NIST to test here.
- **Aspect ratio** non-square pixels - this occurs, it will undermine recognition, but an estimator seems likely to reject wide/narrow faces (how many sigma is acceptable)

Properties to be considered in future

- **Expression neutrality** - we don't have fine-grained expression information such as FACS or classification.
- Localized specular reflections **hot spots** - these should be part of the test, but how to specify severity? As area? Ground truth is not (readily) available.

Software API + implementation

API

- Quality interface and main function call
  - https://github.com/usnistgov/frvt/blob/quality_vector/quality/src/include/frvt_quality.h
- Supporting data types and enumerations
  - https://github.com/usnistgov/frvt/blob/quality_vector/common/src/include/frvt_structs.h

Supporting code

- A toy implementation of the API with random number outputs
  - https://github.com/usnistgov/frvt/blob/quality_vector/quality/src/nullImpl/nullimpl_frvtquality.cpp
- Public validation code, exercising the API
  - https://github.com/usnistgov/frvt/blob/quality_vector/quality/src/testdriver/validate_quality.cpp
  - This code must be executed by developers, and the outputs of the algorithm sent to NIST. NIST will check we can exactly reproduce the outputs on the same input images.
  - We distribute some unusual images (tiny, white, black, textured) in order to stress your code and elicit crashes before you send the code to us. The images are not supposed to represent our main testing images.

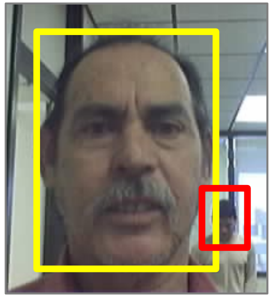
C++ API

```
/**
 * @brief
 * Data structure that stores key-value pairs, with each
 * entry representing a quality element and its value
 */
using QualityAssessments = std::map<QualityItem, double>;
```

```
typedef struct ImageQualityAssessment
{
   FRVT::BoundingBox boundingBox;
   FRVT::QualityAssessments qAssessments;
 };
```

```
typedef struct BoundingBox
{
    int xleft; // leftmost point on head, typically subjects right ear
                // value must be on [0,ImageWidth-1]
    int ytop; // high point of head, typically top of hair
                // value must be on [0,ImageHeight-1]
    int width;    // box width
    int height;     // box height
 };
```

```
/**
  * @brief  This function takes an image and outputs
  * 1) a vector that contains quality items for each
  * 2) location
  * for each face detected in the image
  *
  * @param[in] image
  * Single face image
  *
  * @param[out] assessments
  * A vector of ImageQualityAssessments.  On entry, an empty
  * vector. The implementation should populate it with zero or
  * more entries. Each entry in the vector should contain a tight
  * bounding box and a set of quality values.
  * Each QualityAssessments object should be populated with
  * those quality items that the developer chooses to implement
  */
 virtual FRVT::ReturnStatus
 vectorQuality(
    const FRVT::Image &image,
    std::vector<FRVT::ImageQualityAssessment> &assessments) = 0;
```



```
/** Quality element labels
 */
enum class QualityItem {
  Begin = 0,
  SubjectPoseRoll  = Begin,
  SubjectPosePitch,
  SubjectPoseYaw,
  EyeGlassesPresent,
  SunGlassesPresent,
  Underexposure,
  Overexposure,
  BackgroundUniformity,
  MouthOpen,
  FaceOcclusion,
  Resolution,
  InterocularDistance,
  PixelsFromHeadToLeftEdge,
  PixelsFromHeadToRightEdge,
  PixelsFromChinToBottom,
  PixelsFromHeadToTop,
  ScalarQualityValue,
  End
};
```

- **Elements are options - developers do not need to implement all.**
- Others will be added in future.
- Some may be retracted in the future.

Face count

Task

- Count the number of faces in the image, including those of the subject, people in the background, on T-shirts, in photos on the walls behind, even if cropped.

Motivation

- In applications where one face is assumed, other faces can be detected instead of the intended one, leading to false negatives.

Software output

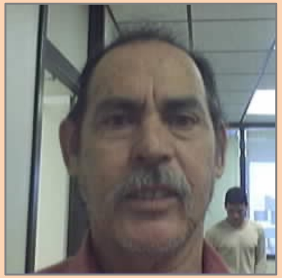
- Instantiate a **FRVT::ImageQualityAssessment** with **FaceBoundingBox** and **QualityElementValues** for each face detected

NIST will execute the code on

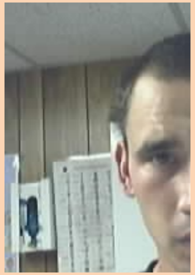
- sets of images with known number of faces, N = 0, 1, 2

NIST will report performance using

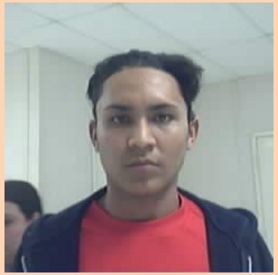
- Statistics on actual vs. reported counts, confusion matrix, overall accuracy
- Tabulate by image type (”wild”, “visa” ...) or conditioned on IOD.


2

0


1

1


2

Non-frontal head orientation

Task

- Estimate the orientation of face (with respect to the camera):
- The head may not be close to the optical axis.

Motivation

- Head orientation other than ISO standard frontal can degrade accuracy

Software output

- Estimates of signed angles in degrees
  - **QualityElement::SubjectPoseRoll**
  - **QualityElement::SubjectPosePitch**
  - **QualityElement::SubjectPoseYaw**

Coordinate system as defined in ISO/IEC 39745-5

NIST will execute the code on images

- with known ground truth orientation (either by-design, or hand-coded)

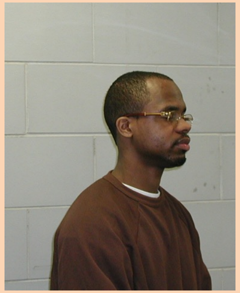
NIST will report performance using

Visualizations of distribution of $\boldsymbol{\theta}_{ESTIMATE}$ and $\boldsymbol{\theta}_{TRUTH}$ and their difference ϕ

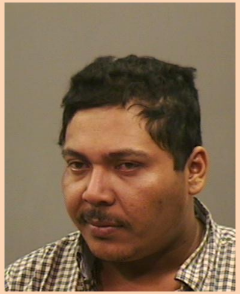
Penalties

- $F_{YAW}(\boldsymbol{\theta}_{ESTIMATE} - \boldsymbol{\theta}_{TRUTH})$
- $F_{PITCH}(\boldsymbol{\theta}_{ESTIMATE} - \boldsymbol{\theta}_{TRUTH})$ tolerant of definitional problem
- $F_{ROLL}(\boldsymbol{\theta}_{ESTIMATE} - \boldsymbol{\theta}_{TRUTH})$

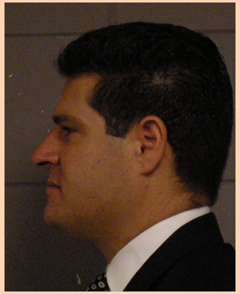
With penalty e.g. $F(\phi) = 1 - \cos(a\phi)$ with scale factor "a" that is more tolerant of pitch angle errors (due to definitional problem), and less tolerant of roll.



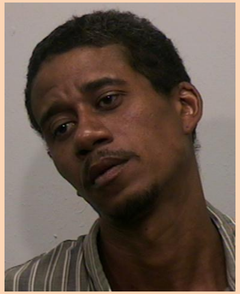
Yaw = +59 degrees
Pitch = 0 degrees
Roll = 0 degrees



Yaw = -37 degrees
Pitch = +4 degrees
Roll = +1 degrees



Yaw = -90 degrees
Pitch = 0 degrees
Roll = 0 degrees



Yaw = -22 degrees
Pitch = +3 degrees
Roll = -18 degrees

Eye glasses present

Task

- Detect transparent eye glasses (but not sunglasses)

Motivation

- Photography specification documents often include a policy for glasses
- False positives from glasses' frames
- False negatives from (change of) glasses

Software output

- Populate **QualityElement::EyeGlassesPresent** with a value on [0,1] giving probability any glasses are present (1.0 for certainty)
- ?Future: Measure frame thickness, make non-dimensional by dividing by estimated interocular distance (IOD)

NIST will execute the code on

- sets of images with and without glasses

NIST will report performance using

- Confusion matrix OR error tradeoff between false negatives (failed detection) and false positive (erroneous detections)
- Summary measure: β FNR + (1-β) FPR with high β

Sunglasses present

Task
- Detect sunglasses (but not transparent eye glasses)

Motivation
- False negatives associated with occlusion of periocular detail
- This element is include separately to eye glasses because policy may dictate different actions for glasses vs. sunglasses

Software output
- Populate **QualityElement::SunGlassesPresent** with a value on [0,1] giving probability sunglasses are present (1.0 for certainty)

NIST will execute the code on
- sets of images with and without sunglasses

NIST will report performance using
- Error tradeoff between false negatives (failed detection) and false positive (erroneous detections)
- Summary measure: β FNR + (1-β) FPR with high β

Mouth open

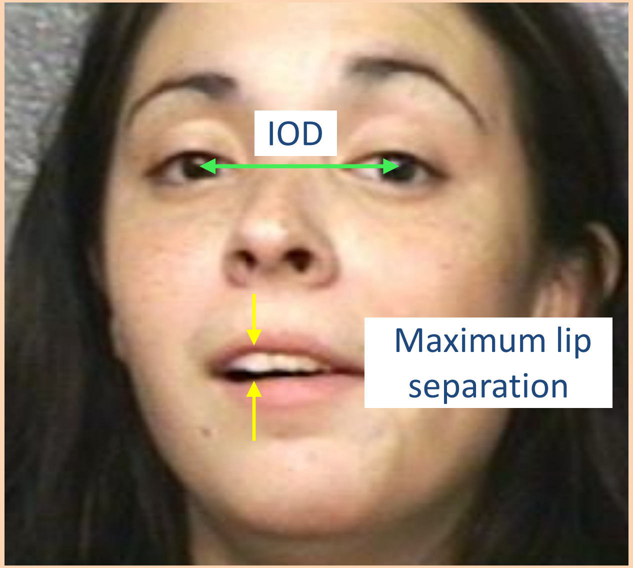
Task
- Determine if the mouth is open, as prohibited in standards
- Normalize lip separation by IOD (necessitating eye-finding)

Motivation
- Possible reduced mate comparison score and increased false negatives due to the change in appearance relative to a reference photo

Software output
- Populate **QualityElement::MouthOpen** with the ratio: measured maximum separation of lips divided by interocular distance (IOD)

NIST will execute the code on
- images with mouth closed
- images with mouth open for which lip separation and IOD are known

NIST will report performance using
- Visualizations of joint distribution of estimated ratio and known ratio



Normalization by IOD as it well-defined and ubiquitously computed. Alternatives such as normalization by lip thickness gives higher fractional error, possible age and ethnicity linkage.

Face occlusion

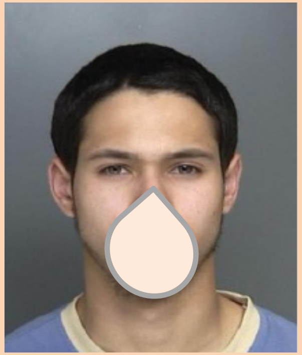
Task
- Quantify the area of the face that is occluded (by objects such as masks, hands, microphones, lecterns)
- Ignore
  - Hair
  - Cropping
  - Sunglasses and eyeglasses

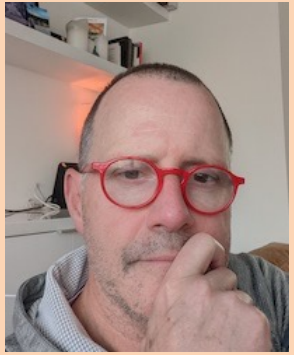
Motivation
- Occlusion can impede detection and elevate FNMR

Software output
- Populate **QualityElement::Occlusion** with proportion of area that is occluded

Evaluation
- Runs on sets of images with various levels of occlusion

NIST will report performance using
- Report pairwise statistics of ground-truth and measured value


33%


28%

0%

Face cropping and margin

Task

- Determine if the face is cropped, or close to the image edge

Motivation

- Cropping can cause detection or recognition failure

Software output: estimate of proximity to edge of image

- **QualityElement::PixelsFromHeadToLeftEdge**
- **QualityElement::PixelsFromHeadToRightEdge**
- **QualityElement::PixelsFromChinToBottom**
- **QualityElement::PixelsFromHeadToTop**
- Negative values when face is cropped, giving estimate of how much is cropped
- Positive values give distance of closest part of the face to the edge

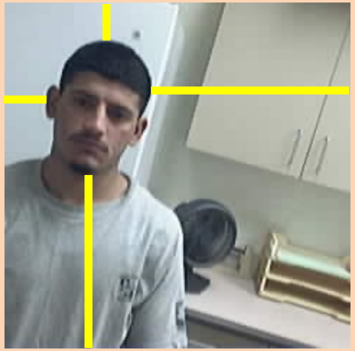
This formulation allows for head rotation, and avoids possible confusion arising from left side of face being in the right hand side of the image.

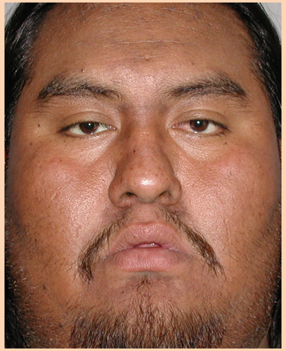
Evaluation

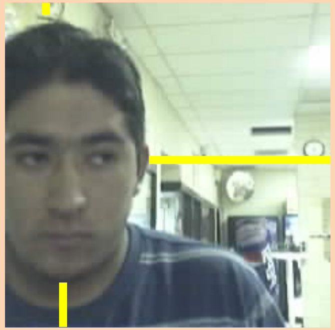
- Runs on sets images with various placements, yaw angles, crops

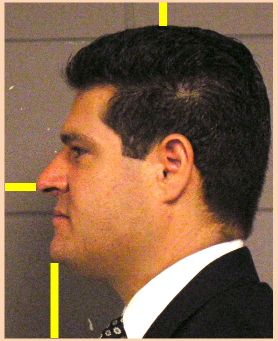
NIST will report performance using

- Report pairwise statistics on estimated vs. ground truth


(30,185,230,36)


(-25,-25,-36,-30)


(-15,105,48,20)


(15,1,48,12)

Background uniformity

Task

- Quantify how uniform the background is

Motivation

- Sufficient illumination non-uniformity will produce false negatives
- Possible false detection (i.e. of other people or non-faces in the background)

Software output

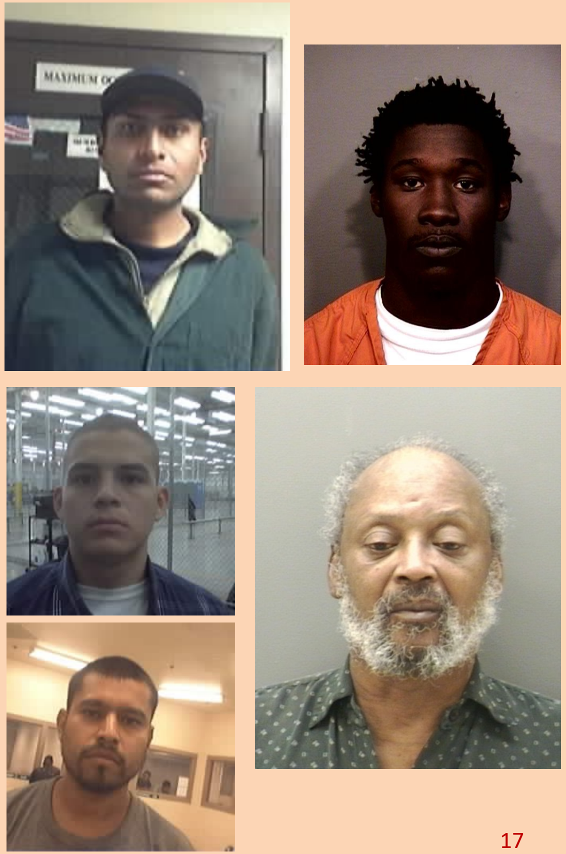
- Populate **QualityElement::BackgroundUniformity** with a value on [0,1] giving degree of non-uniformity of region behind the subject. Higher is worse.

NIST will execute the code on

- With uniform background
- The shadows from the subject head
- With cluttered background

NIST will report performance using

- Some statistics or visualization of actual vs. estimated
- Perhaps an error tradeoff characteristic



Spatial sampling rate

NIST

Task

- Compute the interocular distance (IOD) in pixels
- Use the ISO/IEC {1,3}9794-5 definition
- For images where eyes are not visible due to occlusion or head rotation, produce an IOD estimate based on some (anatomical) model

Motivation

- IOD is a universally understood and widely specified in photography for biometrics, either with a direct value, or implied by the image dimensions (and a known geometry e.g. IOD = W/4)
- Low or high values of IOD are often immediately actionable
- While high IOD is no guarantee of high resolution, low IOD necessarily implies low resolution

Software output

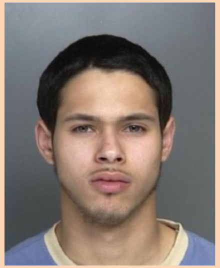
- Populate **QualityElement::InterocularDistance** with a higher is better value on [0,Inf]
- Do not round fractional estimates to integer

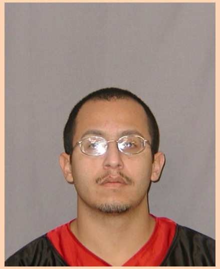
NIST will execute the code on

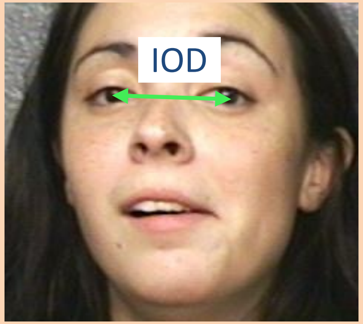
- Frontal images with various estimated IODs.
- Highly non-frontal images (for which we have a frontal image from the same session)

NIST will report performance using

- Error statistics relative to estimated ground truth
- Condition the statistics on IOD and on yaw angle


IOD = 120


IOD = 70



Resolution

Task

- Quantify resolution (blind, without a calibration target). Produce a scalar value that expresses how far from perfect an image is with respect to absence of fine detail of the human face. This factors in all of the following: de-focus, motion blur, compression loss, low spatial sampling rate.
- The software should operate on all images, but should assign highest values to an image with IOD of 256 pixels (close to the EU EES 1376x1024? specification) that is uncompressed, perfectly focused and in all respects pristine. The software should penalize blur more heavily than reduced spatial sampling rate.
- This quantity wraps the four quantities currently drafted in ISO/IEC 29794-5: de-focus, sharpness, motion blur, edge density

Motivation

- Very low resolution gives elevated false negative rates in automated FR, and impedes human review

Software output

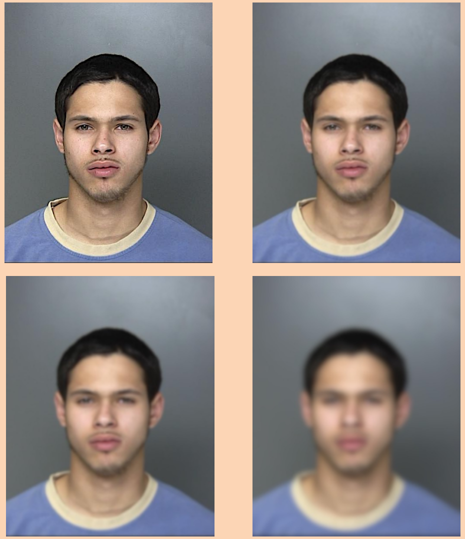
- Populate **QualityElement::Resolution** with a higher-is-better value on [0,1] expressing how much damage is present in an image.

NIST will execute the code on

- sets of images considered to be ideal
- Sets of images with various reductions in resolution applied synthetically and evident in actual images

NIST will report performance using

- Calibration of the measure against genuine similarity scores (?)
- Checks of correct ordering for progressively damaged images.



Underexposure

Task

- Detect underexposure of the face in an image

Motivation

- Under exposure drives higher false negative rates
- Underexposure of ethnicities with lower skin reflectance induces a demographic differential in false negative rates (FNMR, FNIR)

Software output

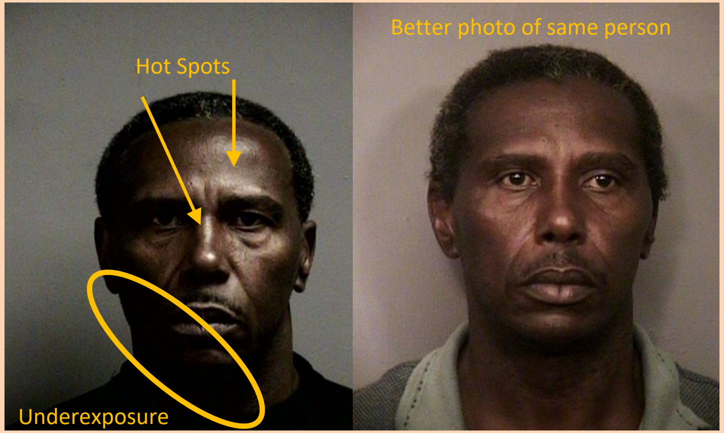
- An **QualityElement::Underexposure** in the face region, higher values are bad

NIST will execute the code on

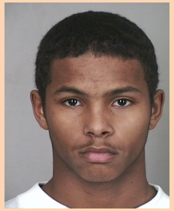
- Hand-selected close-to perfect images and
- Images with a wide range of under-exposure

NIST will report performance using

- Joint distribution measures (QQ plot?) of developer underexposure measure with mated similarity scores produced by several mid-level accuracy FR algorithms comparing the underexposed images with good images.
- Summary statistics (explore rank correlation, partial).



Source: NIST Special Database 32 aka "MEDS", subject S171



NIST's proposal is to related to matching outputs here. The alternative, for NIST to establish an automatically assigned ground-truth measure (e.g. entropy or fraction of area that is "dark"), would lead developers into just re-implementing what NIST did. We seek prediction of continuous mated scores, not binary false negative decisions.

An alternative is for NIST to manually code underexposure, e.g. on 7 point scale.

Overexposure

Task
- Detect overexposure of the face in an image

Motivation
- Overexposure drives higher false negative rates
- Overexposure of ethnicities with high skin reflectance induces a demographic differential in false negative rates (FNMR, FNIR)

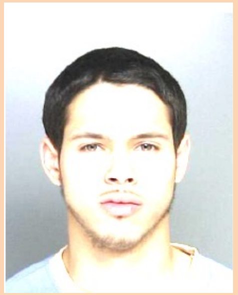
Software output
- An **QualityElement::Overexposure** in the face region, higher values are bad

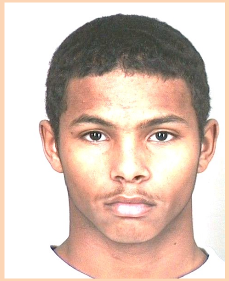
NIST will execute the code on
- Hand-selected close-to perfect images and
- Images with a wide range of overexposure

NIST will report performance using
- Joint distribution measures of developer overexposure measure with mated similarity scores produced by several mid-level accuracy FR algorithms comparing the overexposed images with good images.
- Summary statistics (explore rank correlation, partial).

NIST’s proposal is to related to matching outputs here. The alternative, for NIST to establish a ground-truth measure (e.g. entropy or fraction of area that is “light”), would lead developers into just re-implementing what NIST did. We seek prediction of continuous mated scores, particularly low scores, not binary false negative decisions.

An alternative is to manually code overexposure, e.g. on 7 point scale.


Source: NIST Special Database 32 aka “MEDS”
Modified in powerpoint.


Source: NIST Special Database 32 aka “MEDS”
Modified in powerpoint.

Scalar quality value

Task

- Summarize utility of an image for recognition.

Motivation

- Various use-cases seek a single number that can be thresholded for yes/no acceptance decisions, or used to summarize quality over some large collections

Software output

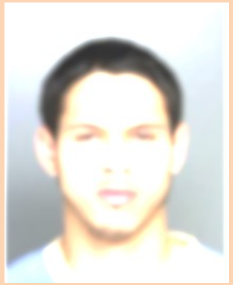
- An **QualityElement::ScalarQualityValue** higher values are good

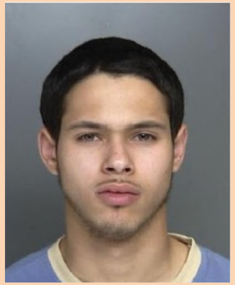
NIST will execute the code on

- Images that yield false negatives when compared with ISO-like reference images
- Images that do not yield false negatives

NIST will report performance using

- Statistics that associate low quality with higher likelihood of FNMR, including FNMR vs. Q; FNMR vs low Q rejection proportion; relationship of Q values and mated comparison scores.


Q = 30


Q = 97